User Controls with Visual C#

ما هي الـ User Control؟

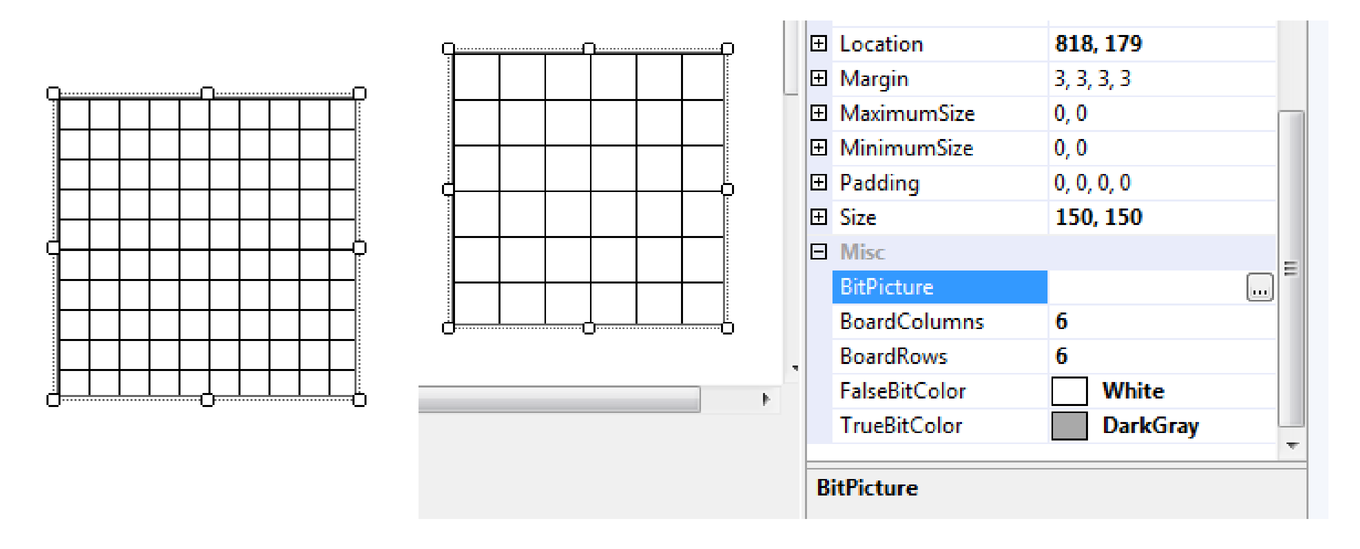
هي عبارة عن component له تمثيل مرئي مثل (Button, Lable, PictureBox, …etc)، يمكن إنشاءه وإعادة استخدامه كأي أداة أخرى من أدوات الـ Visual Studio الموجودة ضمن الـ ToolBox، ومن أنواعه:

- **Extended Controls** وفيها لدينا component جاهز قمنا بتطويره عن طريق الوراثة مثلاً، وكمثال على هذا كأن نقوم بإضافة مزايا وخصائص جديدة إلى الـ PictureBox كخاصية ضبط طول وعرض الصورة بحيث تكون نسبة طوله إلى عرضه صحيحة ومتوضع ضمن الـ PictureBox.

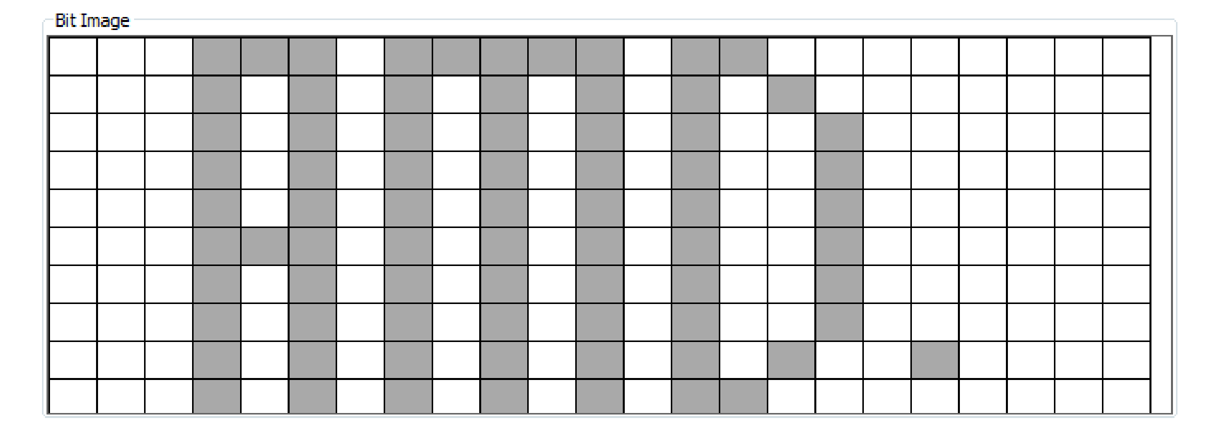

PictureBox


NewPictureBox

مثال آخر وهو ما قمت به فعلاً، component يمثل PictureBox شبكية؟ وفيها تقسم إلى عدة أسطر وإلى عدة أعمدة عن طريق رسم خطوط فقط، وهناك أحداث تقوم بتحليل ضغطات زر الفأرة للمستخدم إلى indices لمعرفة سطر وعمود المربع الذي ضغط عليه فيتم تلوينه بلون ما يمكن تغييره، حيث يمكنه بالزر الأيمن المسح وبالأيسر الكتابة.



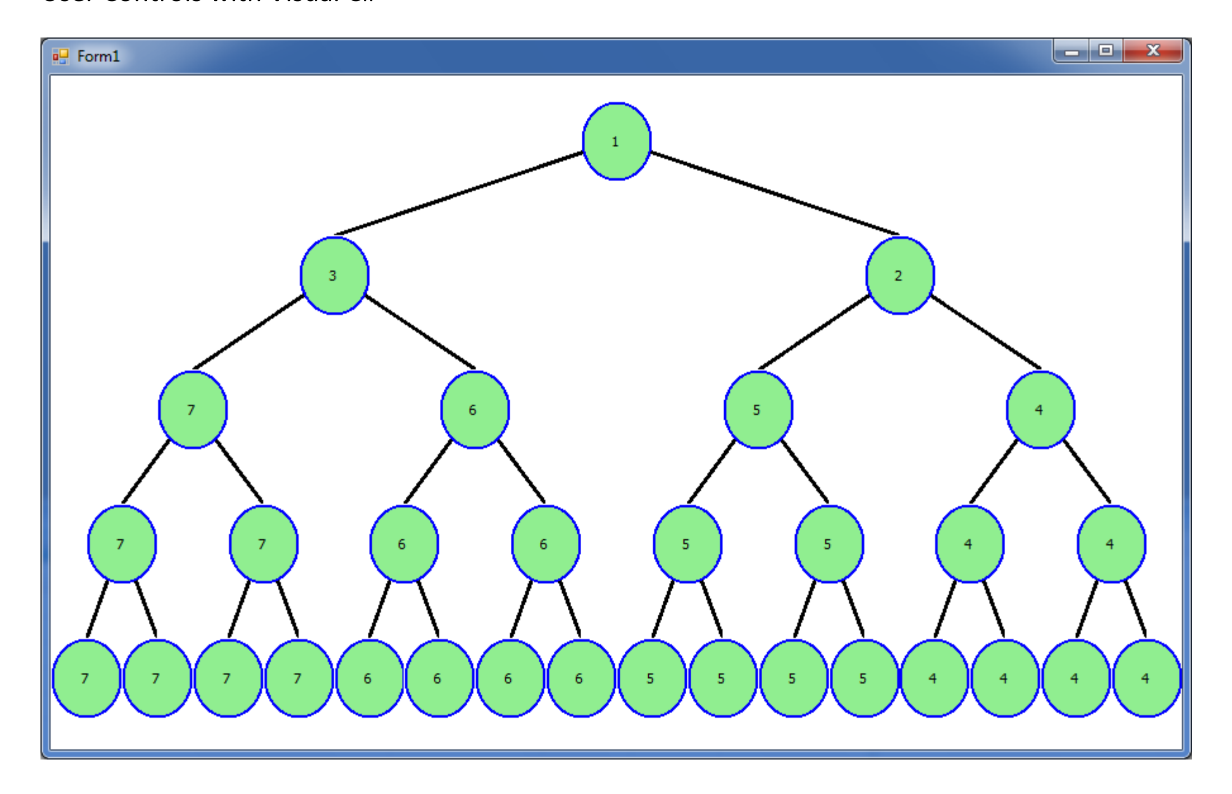
إعداد الطالب: محمد سياف



- **Composite Controls** وفيها يتم جمع أو ضم component أو أكثر لتصميم component جديد، وكمثال على ذلك أن نقوم بتصميم component يمثل مستعرض صور (بصراحة component ضخم إلى حد ما ☺)، وفيه سيتم وضع PictureBox وبضعة Buttons حسب الحاجة للانتقال إلى الصورة التالية والسابقة و...إلخ ويمكن إضافة Button للاستعراض و Buttons أخرى لقلب أو تكبير أو حفظ الصورة ☺، ومن الممكن أن نقوم بإضافة خاصية تدعى AutoSlide عندها يقوم تلقائياً بعرض الصورة تلو الصورة وكأنها تتحرك ضمن الـ PictureBox ☺.

مثال آخر: أن نقوم بتصميم component يمثل Lable ولكن بعد إضافة بعض الخصائص والتوابع وليكن اسمه الجديد SmartLable ☺ (لا تدققوا)، وفيه خاصية جديدة أو حدث إن جاز التعبير هو حدث DoubleClick الذي سنقوم فيه بالتالي: إخفاء الـ Lable وإنشاء TextBoxعوضاً عنه برمجياً ☺، ما الفائدة؟ الفائدة أن نقوم بإعادة تسمية الـ Lable وعند ضغط Enter يختفي الـ TextBox ويظهر الـ Lable من جديد بالاسم الجديد ☺.

- **Entirely a new control** أما هنا نقوم بإنشاء UserControl جديد تماماً قد يحتوي على component جاهز وقد لا يحتوي، مثال: إنشاء component يمثل شجرة (بالخورازميات 2 كما نعلم)، وهو فعلا ما قمت به ضمن وظيفة الخوارزميات 2، وفيه قد تم رسم دائرة أستطيع التحكم بألوانها وضمنها يوجد Lable أستفيد منه بوضع قيمة ضمن الدائرة، وبالطبع يوجد العديد من الأحداث المهمة كحدث تغير حجم الـ UserControl الذي يقوم بإعادة رسم الدائرة، أو حدث إضاءة عقدة الذي يقوم بتغيير لون عقدة من خلال إعادة رسمها لمدة معينة ثم يقوم بتغيير اللون من جديد ويعيد رسمها ...إلخ.



هذا الشكل بكاملة (طبعا عدا الـ Form) UserControl تدعى Tree تتألف من اثنتين من الـ UserControls الأولى تمثل الخط وتدعى DLine والثانية تمثل العقدة وتدعى DDNode، أما الـ Tree هي الشجرة التي تقوم بالتحكم والإنشاء الديناميكي لكل منهم حسب الحاجة.

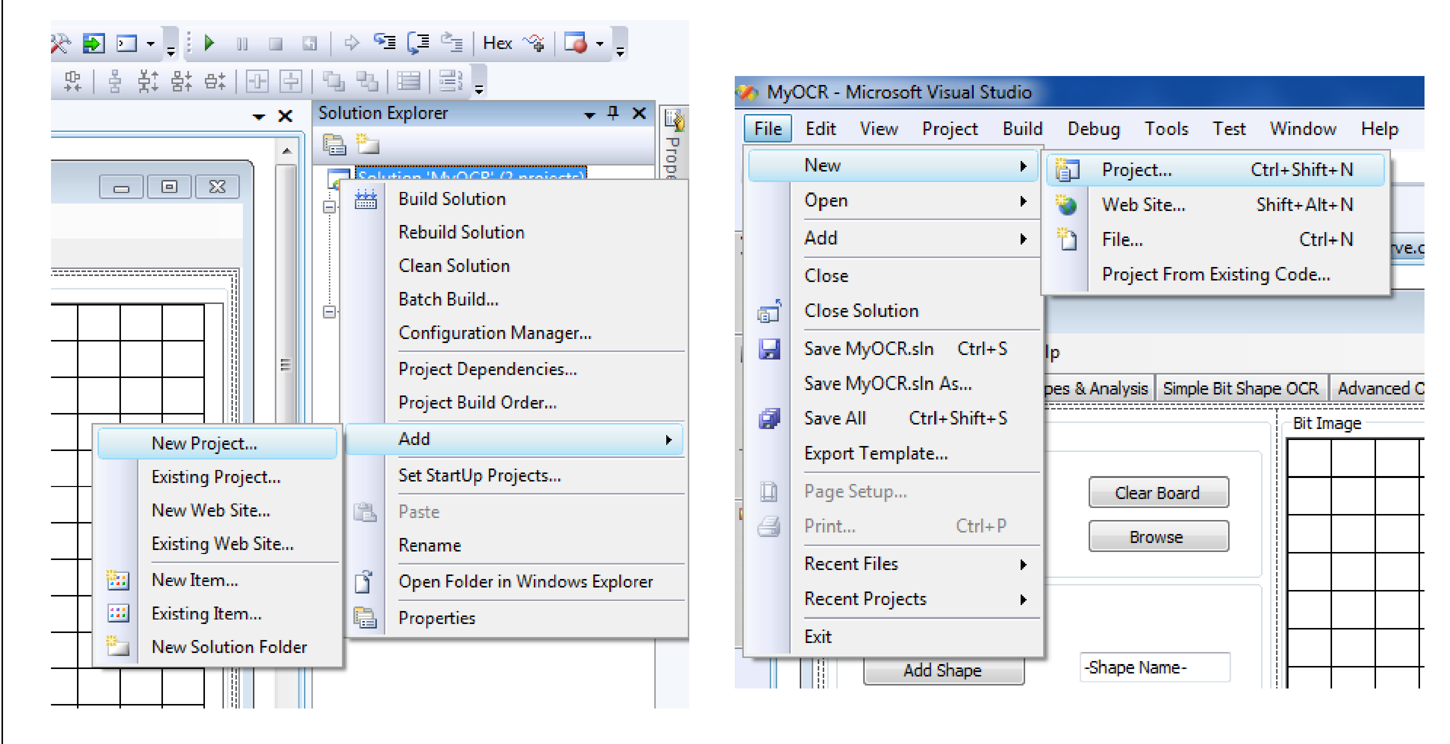
ما فائدة الـ User Controls؟

- ✓ إمكانية تطوير components موجودة لنحقق بعض الأهداف والمكاسب الشخصية ☺.
- ✓ إمكانية إنشاء components جديدة لها معاني جديدة قد تكون لها علاقة بالخورزميات مثلاً (شجرة كما رأينا قبل قليل)، وذلك بالإعتماد على components أخرى أو بدونها.
- ✓ إمكانية إعادة استخدام وتجديد ما تم إنشاءه أو تطويره.
- ✓ فوائد ليست في ذهني الآن، وأخرى لا علم لنا بها ☺.

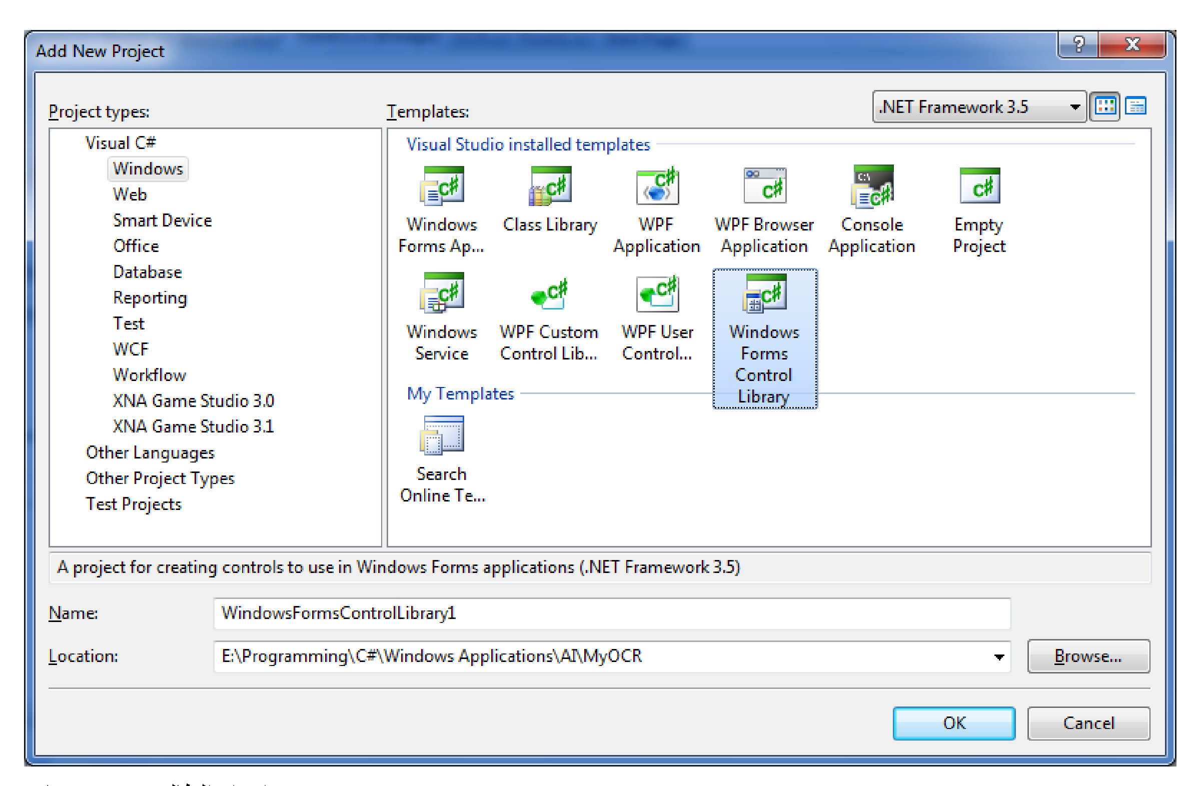
كيف يمكن إنشاء User Control؟

لن أخبركم ☺، يتم عن طريق إنشاء مشروع جديد ... و ....، من الأفضل أن نتابع الصور ☺:

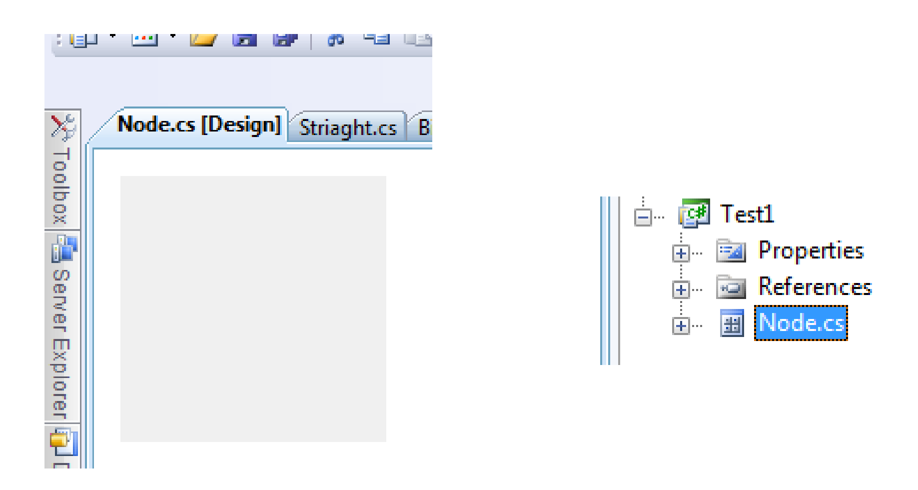
1) نقوم بإضافة مشروع جديد إما ضمن مشروعنا الحالي ويكون بضغط الزر الأيمن فوق Solution المشروع فنختار Add ومن ثم New Project، أو عن طريق قائمة File ثم New ثم Project.



2) عندها نقوم باختيار نوع المشروع " Windows Forms Control Library"، ونقوم باختيار أي اسم للمشروع.



3) نلاحظ ظهور المشروع ضمن Solution Explorer بعد أن سميته باسم " Test1" و الـ UserControl بـ Node، حيث يمكن تحرير أو تغيير اسم الـ UserControl أو أياً كان ضمن Solution Explorer وذلك بالضغط عليه مرتين ومن ثم تغيير الاسم.



ونلاحظ أيضاً نافذة التصميم الخاصة بال UserControl Node، وفيها عبارة عن Panel صغيره يمكن التحكم بأبعادها أو قصها ...إلخ.

4) سنقوم بتصميم عقيدة بسيطة ☺، سنقوم بداية برسم دائرة، ومن الجدير بالذكر أن الرسم يمكن أن يكون على Panel أو Button أو PictureBox ...إلخ، ومن الجدير بالذكر أيضاً ☺ أن الرسم على الـ components السابقة ليس بثابت أقصد أنه عند وجود tab آخر في البرنامج، عند الانتقال إليه والعودة نجد أن الرسم اختفى والحال نفسها في حال قمنا بتصغير البرنامج أو بفتح برنامج آخر فوقه أو إذا علق البرنامج.

من أجل المشكلة السابقة نقوم باستخدام ما يسمى بالـ ☺ Bitmap، من هذا أيضاً؟ صف أو class يمكننا من التعامل مع الصور (تخزين، تعديل، حفظ ...) والأهم أنه يمكننا الرسم عليه دون أن يختفي أي جزء من الشكل (أمر طبيعي لأن الصورة أصبحت مخزنة)، بعد هذه المقدمة الطويلة سأريكم كود رسم دائرة.

File Edit View Refactor Project Build Debug Data Tools Test Window Help

Debug | Any CPU | GetCur

Start Page | Node.Designer.cs | Node.cs | Node.cs [Design] | Striaght.cs | BinaryFiles.cs | BitImage.cs | Program.cs | Shape.cs | BasicCurve.cs | Form1.cs [Desig

Toolbox | Server Explorer | Data Sources

Test1.Node | g

```
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Data;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Test1
{
    public partial class Node : UserControl
    {
        Graphics g;
        Bitmap bitmap;

        public Node()
        {
            InitializeComponent();
            DrawCircle();
        }

        public void DrawCircle()
        {
            bitmap = new Bitmap(this.Width, this.Height);
            g = Graphics.FromImage(bitmap);
            g.DrawEllipse(new Pen(Brushes.Black,3), new Rectangle(5,5,this.Width-10, this.Height-10));
            this.BackgroundImage = bitmap;
        }
    }
}
```

قمنا هنا بتعريف متحول من نوع Bitmap وحددنا له أبعاد الصورة (وهي نفس حجم الـ UserControl) ثم تم تعريف متحول من صنف Graphics وحددنا له أين يجب عليه الرسم، ومن ثم قمنا برسم دائرة (وهي حالة خاصة) وإسناد الصورة المحتواة في الـ Bitmap إلى الخلفية ☺.

5) أما شكل الـ Node أصبح (وضوحاً ☺):



6) سنقوم بتطوير ما سبق (بوكيمون ☹): سنقوم بجعل حجم الدائرة بحجم الـ Node تماماً بدل القيم الابتدائية (كون التابع لن يستدعى سوى مرة واحدة) هذا أولاً ثانياً نريد أن نراها فيجولياً ☺، لذلك نقوم بإضافة حدث تغير حجم الـ Node ونكتب فيه:

```
private void Node_SizeChanged(object sender, EventArgs e)
{
    DrawCircle();
}
```

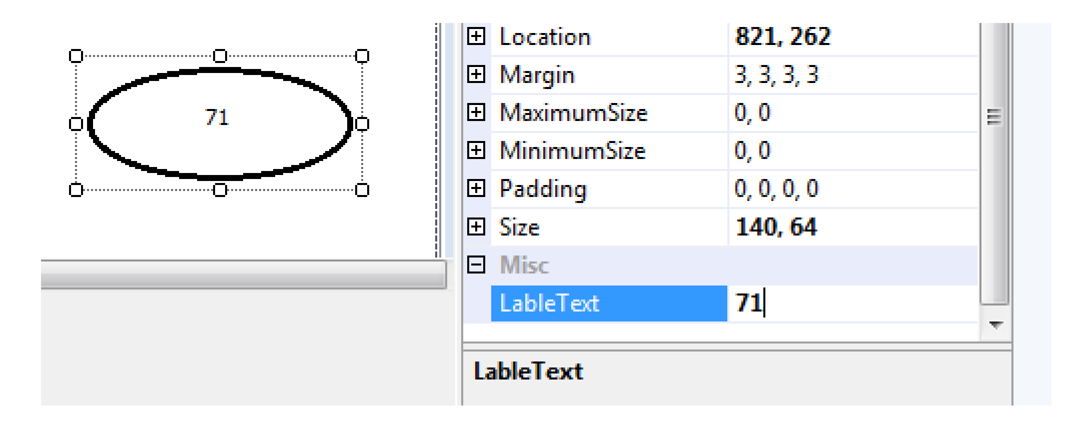
الآن عند استخدام الـ Node يمكن رؤية الدائرة المرسومة وعند تغيير حجم الـ Node نلاحظ تغير حجم الدائرة.

7) نكمل عملية تصميم الـ Node، الآن نقوم بإضافة Lable إلى الـ Node، ونضيف سطراً إلى الكود السابق مفاده تغيير موقع الـ Lable ليصبح بمنتصف الدائرة، ولكي نقوم بتغيير الـ Text الخاص بالـ Lable نقوم بإضافة Property معينة كالتالي:

```
        this.BackgroundImage = bitmap;
        label1.Location = new Point(this.Width/2-10, this.Height/2-10);
    }

    public String LableText
    {
        get { return label1.Text; }
        set { label1.Text = value; }
    }
```

عند كتابة الكود السابق نجد الخاصية السابقة ضمن خصائص الـ Node في قائمة الـ Properties التي تحوي جميع خصائص هذا الـ component.



8) لكم هنا كامل الحرية في إكمال المثال السابق ☺.

وهنا تنتهي رحلتنا مع هذا المثال.

ملاحظة على الطريق: يمكن قص الـ Panel لتصميم الشكل الذي نريده، كيف؟

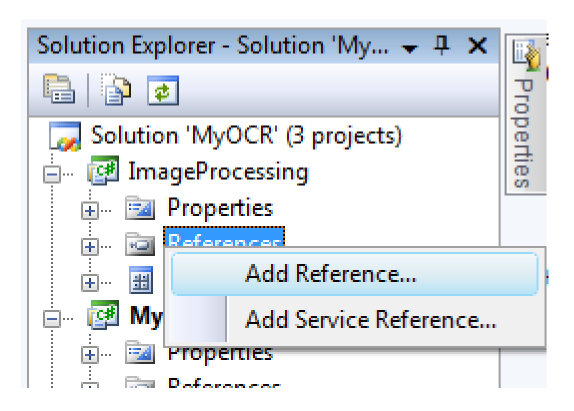
```
    }
    System.Drawing.Drawing2D.GraphicsPath gp;
    gp = new System.Drawing.Drawing2D.GraphicsPath();
    gp.AddEllipse(0, 0, panel1.Width, panel1.Height);
    this.Region = new Region(gp);
}
```

☹ السكوت علامة الرضا، الكود واضح من عنوانه ☺، صنف الـ GraphicsPath هو صنف ☹ نستخدمه لتخزين النقاط بشكل مختصر جداً، ولكي نقوم بقص أي شكل يجب أن نرسم الشكل الذي نريد (قد يكون تتالي من المستقيمات أو النقاط التي تحدد بالنهاية منحني مغلق)، الصف السابق يمكننا من ذلك ☺ فمثلا أريد شكل الـ UserControl على شكل دائرة عندها أخزن الدائرة (التي تمثل حدود الشكل) ضمن متحول من الصف السابق، ومن ثم نقوم بقص الشكل بحسب حدود الدائرة التي رسمتها من خلال التعليمة السابقة.

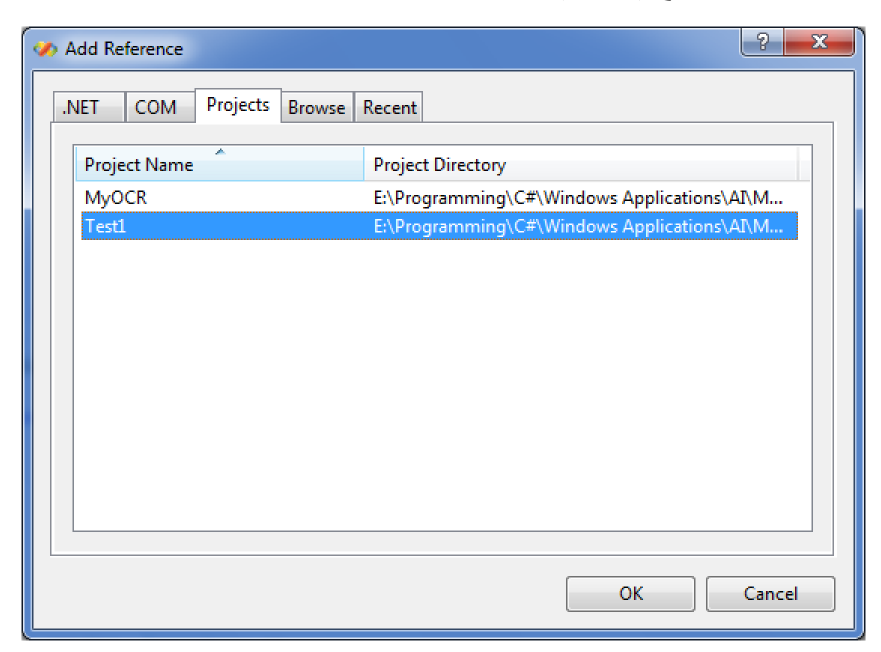
كيف يمكن استخدام أو تضمين User Control؟

أظنكم كنتم تريدون معرفة كيف يتم ذلك ☺، يتم ذلك باتباع الخطوات التالية:

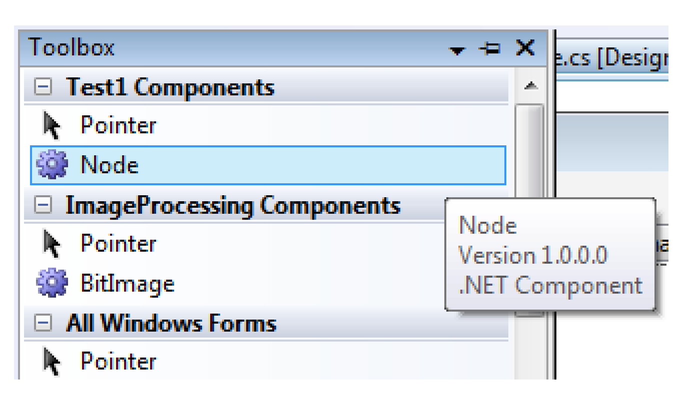
1) نضغط بالزر الأيمن فوق الـ References ونختار Add Reference.



2) ونفتح الـ tab الـ Projects في حال كان قمنا بإضافته مشروع UserControl إلى مشروعنا الأصلي، أو نختار Browse لنختار المشروع (Test1) الذي صممنا فيه الـ Node:



3) عندها سنجد مشروعنا السابق كـ dll ضمن الـ References، الخطوة التالية (وهي خفية نوعاً ما) أن نقوم بتنفيذ المشروع أقصد (Control + F5)، عندها سيظهر ضمن الأدوات (ToolBox) مجموعة أدوات جديدة باسم "Test1 Components" نستطيع أن نختار منها الـ component الذي صممناه واستخدامه .☺



في حال كان المشروع " Test1" موجوداً ضمن المشروع الأصلي، والمشروع الأصلي يستخدم الـ component "Node"، يجب أن نقوم بإعادة الـ Debugging أي بضغط (Control + F5) عندها ستتفعل التغيرات التي قمنا بها في المشروع "Test1".

ملاحظة على الهامش: يمكن تحويل الـ component السابق إلى dll (حتى ولو كان عبارة عن مجموعة UserControls بعدة ملفات) واستخدامه ضمن أي مشروع، ولكني لا أذكر الآن التعليمة التي تقوم بذلك، وبكل الأحوال لا أنصحكم بذلك مالم تقوموا بتصميم component كامل مع معالجة مختلف جميع أنواع الـ Exceptions التي من الممكن أن ترد وحتى التي لن ترد ☺ (اسئلو المجرب)، لأنه عند ورود أي Exception عند استخدام الـ dll، لن يتم معرفة مشكلته وسنضطر عندها للرجوع إلى مشروعه الأصلي واكتشاف متى سيظهر هذا الخطأ معنا ثم نحاول معرفة أسبابه ومعالجته.

يمكن إنشاء components لوظائف الأوتومات أو الخوارزميات أو المشاريع.

ونهاية نشكر حسن استماعكم، ونرجو أن تعم الفائدة عليكم وعلينا، والحمد لله...

وأعتذر عن ورود أي خطأ، ولا مشكلة في إعلامي بها إن وردت...

لا تنسونا من صالح الدعاء...